

Claims

What is claimed is:

- 1 ✓ 1. A method of priority queue dispatching in a data processing system,
2 wherein:
3 a critical section of code for a task is when the task has at least one of
4 a plurality of locks locked; and
5 said method comprises:
6 A) dispatching a first task for execution at a temporary dispatching
7 priority higher than a standard dispatching priority for that first
8 task whenever the first task is in a critical section of code for
9 that first task, wherein the critical section of code for that first
10 task is a result of that first task having locked a first lock
11 without having unlocked the first lock; and
12 B) dispatching the first task for execution at the standard dispatching
13 priority for that first task whenever the first task is not in the
14 critical section of code for that first task.
- 1 2. The method in claim 1 wherein:
2 the method further comprises:
3 C) queuing a second task in a first FIFO queue when the second task
4 attempts to lock the first lock and fails, wherein:
5 the FIFO queue comprises a set of tasks waiting to lock the first
6 lock; and
7 step (C) comprises:
8 1) placing the second task at an end of the first FIFO queue;
9 2) comparing the temporary dispatching priority for a third
10 task that is ahead of the second task in the FIFO queue to
11 the temporary dispatching priority of the second task; and
12 3) setting the temporary dispatching priority of the third
13 task to the temporary dispatching priority of the second
14 task when the temporary dispatching priority of the
15 second task is determined in substep (2) to be greater
16 than the temporary dispatching priority of the third task.

1 3. The method in claim 2 wherein:
2 step (C) further comprises:
3 4) setting the temporary dispatching priority for the second task to the
4 standard dispatching priority for the second task before substep
5 (2) and (3).

1 4. The method in claim 1 wherein:
2 the method further comprises:
3 C) queuing a second task in a first FIFO queue when the second task
4 attempts to lock the first lock and fails, wherein:
5 the FIFO queue comprises a set of tasks waiting to lock the first
6 lock; and
7 step (C) comprises:
8 1) placing the second task at an end of the first FIFO queue;
9 2) comparing the temporary dispatching priority for the first
10 task to the temporary dispatching priority of the second
11 task; and
12 3) setting the temporary dispatching priority of the first task
13 to the temporary dispatching priority of the second task
14 when the temporary dispatching priority of the second
15 task is determined in substep (2) to be greater than the
16 temporary dispatching priority of the first task.

- 1 5. The method in claim 4 wherein step (C) further comprises:
2 4) testing whether the first task is in a second FIFO queue awaiting a
3 chance to lock a second lock; and
4 5) upgrading the temporary dispatching priority of a third task that
5 has a second lock locked when the first task is determined in
6 substep (4) to be in the second FIFO queue awaiting a chance to
7 lock the second lock, wherein substep (5) comprises:
8 a) comparing the temporary dispatching priority for the
9 third task to the temporary dispatching priority of the
10 second task; and
11 b) setting the temporary dispatching priority of the third
12 task to the temporary dispatching priority of the second
13 task when the temporary dispatching priority of the
14 second task is determined to be greater than the
15 temporary dispatching priority of the third task.
- 1 6. The method in claim 5 wherein:
2 the method further comprises:
3 D) comparing a highest temporary dispatching priority of any task in
4 the first FIFO queue to the temporary dispatching priority of the
5 first task after the first task unlocks the second lock; and
6 E) setting the temporary dispatching priority of the first task to the
7 highest temporary dispatching priority of any task in the first
8 FIFO queue when the temporary dispatching priority of the first
9 task is determined in step (D) to be greater than the highest
10 temporary dispatching priority of any task in the first FIFO
11 queue.

1 13. The software in claim 12 wherein:
2 set (C) further comprises:
3 4) setting the temporary dispatching priority for the second task to the
4 standard dispatching priority for the second task before subset
5 (2) and (3).

1 14. The software in claim 11 wherein:
2 the software further comprises:
3 C) queuing a second task in a first FIFO queue when the second task
4 attempts to lock the first lock and fails, wherein:
5 the FIFO queue comprises a set of tasks waiting to lock the first
6 lock; and
7 set (C) comprises:
8 1) placing the second task at an end of the first FIFO queue;
9 2) comparing the temporary dispatching priority for the first
10 task to the temporary dispatching priority of the second
11 task; and
12 3) setting the temporary dispatching priority of the first task
13 to the temporary dispatching priority of the second task
14 when the temporary dispatching priority of the second
15 task is determined in subset (2) to be greater than the
16 temporary dispatching priority of the first task.

1 15. The software in claim 14 wherein set (C) further comprises:
2 4) testing whether the first task is in a second FIFO queue awaiting a
3 chance to lock a second lock; and
4 5) upgrading the temporary dispatching priority of a third task that
5 has a second lock locked when the first task is determined in
6 subset (4) to be in the second FIFO queue awaiting a chance to
7 lock the second lock, wherein subset (5) comprises:
8 a) comparing the temporary dispatching priority for the
9 third task to the temporary dispatching priority of the
10 second task; and
11 b) setting the temporary dispatching priority of the third
12 task to the temporary dispatching priority of the second
13 task when the temporary dispatching priority of the
14 second task is determined to be greater than the
15 temporary dispatching priority of the third task.

1 16. The software in claim 15 wherein:
2 the software further comprises:
3 D) comparing a highest temporary dispatching priority of any task in
4 the first FIFO queue to the temporary dispatching priority of the
5 first task after the first task unlocks the second lock; and
6 E) setting the temporary dispatching priority of the first task to the
7 highest temporary dispatching priority of any task in the first
8 FIFO queue when the temporary dispatching priority of the first
9 task is determined in set (D) to be greater than the highest
10 temporary dispatching priority of any task in the first FIFO
11 queue.

00333T-4040450

- 1 / 21. A computer readable Non-Volatile Storage Medium encoded with
2 software for priority queue dispatching in a data processing system,
3 wherein:
4 a critical section of code for a task is when the task has at least one of
5 a plurality of locks locked; and
6 said software comprises:
7 A) dispatching a first task for execution at a temporary dispatching
8 priority higher than a standard dispatching priority for that first
9 task whenever the first task is in a critical section of code for
10 that first task, wherein the critical section of code for that first
11 task is a result of that first task having locked a first lock
12 without having unlocked the first lock; and
13 B) dispatching the first task for execution at the standard dispatching
14 priority for that first task whenever the first task is not in the
15 critical section of code for that first task.